AMENDMENTS TO CLAIMS

1. (Currently amended) A method of switching context on a processor, the method comprising:

saving the context under software control using an inconsequential register; [[and]]

preventing the processor from changing the context while the context is being saved; and

thereafter restoring the context using an inconsequential register.

2. (Original) The method of claim 1, wherein the inconsequential register is used as a temporary storage in lieu of a privileged register.

3. (Original) The method of claim 1, wherein the context is saved at a predetermined interruption point.

4. (Original) The method of claim 1, wherein the context is switched between a host operating system and a virtual machine application, the virtual machine application controlling the context switch.

5. (Original) The method of claim 4, wherein the inconsequential register is used to pass information to the virtual machine application.

6. (Original) The method of claim 1, wherein the context is switched using an IA-64 processor.

7. (Previously presented) The method of claim 6, wherein the inconsequential register is a caller-save register.

8. (Previously presented) The method of claim 6, wherein the inconsequential register is a branch register.

9. (Cancelled)

10. (Currently amended) The method of claim [[9]] 1, wherein the context is restored by using a branch register to perform an indirect branch.

11. (Currently amended) A method of switching context between a host OS and a virtual machine on a processor, the processor having privileged registers, the processor having access to other memory, the method comprising:

giving the virtual machine access to the privileged registers;

using at least one privileged register as temporary storage to save the context in the other memory at a predetermined interruption point; [[and]]

preventing the processor from changing the context while the context is being saved; and

restoring the context using an inconsequential register;

the virtual machine application controlling the context switch.

12. (Currently amended) Apparatus comprising:

a processor including a plurality of registers; and

a virtual machine application for commanding the processor to switch context by saving the context under software control using an inconsequential register of the processor as temporary storage; and preventing the processor from changing the context while the context is being saved; and to restore context by using the inconsequential register.

13. (Original) The apparatus of claim 12, wherein the inconsequential register is used as a temporary storage in lieu of a privileged register.

14. (Original) The apparatus of claim 12, wherein the context is saved at a predetermined interruption point.

15. (Original) The apparatus of claim 12, further comprising a host OS; wherein the context is switched between the host OS and the virtual machine application; and wherein the virtual machine application controls the context switch.

16. (Original) The apparatus of claim 15, wherein the inconsequential register is used to pass information to the virtual machine application.

17. (Original) The apparatus of claim 12, wherein the processor is an IA-64 processor.

18. (Previously presented) The apparatus of claim 17, wherein the inconsequential register is a caller-save register.

19. (Previously presented) The apparatus of claim 17, wherein the inconsequential register is a branch register.

20. (Currently amended) The apparatus of claim 12, wherein the virtual machine application [[further]] commands the processor to restore the context using the inconsequential register as temporary storage.

21. (Original) The apparatus of claim 20, wherein the context is restored by using a branch register to perform an indirect branch.

22. (Currently amended) An article comprising computer memory encoded with ~~Software comprising~~ instructions for commanding a processor to switch context by saving the context under software control using an inconsequential register of the processor as temporary storage; and preventing

the processor from changing the context while the context is being saved; the instructions further commanding the processor to restore context using the inconsequential register.

23. (Previously presented) The method of claim 1, wherein content of the inconsequential register is corrupted during the context switch.

24. (Previously presented) The method of claim 1, wherein using the inconsequential register includes storing an address in the inconsequential register, the address indicating a memory location at which the context will be saved.

25. (Previously presented) The method of claim 1, wherein the inconsequential register does not store context at a predetermined interruption point.

26. (Previously presented) The method of claim 1, wherein the context is stored in memory other than the inconsequential register.